# REMARKS

In the Office Action, the Examiner rejected the claims under 35 USC §102. These rejections are fully traversed below. Claims 1-24 remain pending.

Reconsideration of the application is respectfully requested based on the following remarks.

## REJECTION OF CLAIMS UNDER 35 USC §102

In the Office Action, the Examiner rejected claims 1-22 under 35 USC §102 as being anticipated by Preisler et al, U.S. Patent No. 5,675,803, ('Preisler' hereinafter). This rejection is fully traversed below.

In order for a claim to be anticipated under 35 USC §102, the cited reference must teach every aspect of the claimed invention. Applicant respectfully submits that Preisler neither discloses nor suggests each of the claimed features, as will be set forth in further detail below.

Preisler relates to a method and apparatus for a fast debugger fix and continue operation. See title. A user can begin a debugging session, and if an error in the code is encountered, the user can edit the corresponding source code to correct the error and then execute a "Fix and Continue" command all without leaving the debugging session. The Fix and Continue code calls the compiler to recompile the source code file with the edited text in it, receives the resulting compiled object code file from the compiler, uses the dynamic linker to link the recompiled object code into the target application process, patches the previous version of this same object code file to refer to the newly recompiled code, resets any required variables and registers, and resets the program counter to the line of code being executed when the error was discovered. See Abstract.

Specifically, Preisler discloses a debugger in which locations that need to be patched are identified as patch sites. The original instructions at these patch sites are then replaced by a branch to patch area instruction. See Col. 5, lines 60-65. As set forth above, when an error is detected during a debugging session, the user must edit the source code to correct the error and initiate the "Fix and Continue" command, which recompiles the modified code and uses

the dynamic linker to patch the previous version of the object code file. Thus, the user may dynamically fix program code which contains an error without terminating a debugging session. Stated another way, <u>code cannot be modified without user intervention.</u>

It is also important to note that Preisler relates to a <u>debugger.</u> The debugger is used by a user to debug source code. During a debug session, the user fixes source code, which is then compiled and linked. Specifically, the debugger program identifies the patch sites, and accumulates information such as whether a particular patch site should be patched and the size of memory being accessed. The debugger then takes the list of patch area section sizes and attempts to allocate space for them. See Col. 6, lines 27-64.

In contrast, the presently claimed invention is <u>not directed to a debugger,</u> but a method and system for performing a system function such as I/O functionality. This is accomplished through a set of primitive functions corresponding to a set of system services. The primitive functions, although "equivalent" to the set of system services, are reduced in functionality as well as performance. In this manner, a secondary mechanism for performing the set of system services is made available in the event of a system error. Thus, the presently claimed invention may be used to ensure functionality of system services such as I/O services required to debug a system successfully.

It is also important to note that through the presently claimed invention, a system service may be replaced by a primitive function having equivalent, but reduced functionality and/or performance. Preisler <u>neither discloses nor suggests replacing a system service with a function having reduced functionality.</u> In fact, by the very nature of debugging, a user modifying a set of source code will generally increase the source code's functionality.

With respect to claim 1, Preisler neither discloses nor suggests a method or computer-readable medium providing replacement functions for a set of system services including "sending a request for a primitive function from one of the set of system services to another one of the set of system services, the primitive function replicating the one of the set of system services in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error." In fact, Preisler neither discloses nor suggests such <u>communication between two system services,</u> as claimed. Rather, a debugger operates to link and therefore replace code with a modified, compiled version. More specifically, Preisler fails to disclose or suggest a <u>request from a first system service to a second system service for</u>

a primitive function replicating the first system service. The Examiner alleges that Preisler discloses the claimed features. However, the cited portion of Preisler states that "the original instructions at these patch sites are then replaced by a branch to patch area instruction." See Col. 5, lines 60-65. In other words, Preisler discloses a debugger that effectively replaces an original set of instructions with a modified (e.g., corrected) set of instructions. However, a patch instruction merely enables a section of code to be called or "patched." It is important to note that the code being replaced in Preisler does not communicate with the new section of code, nor does the code being replaced send a request for code that replicates the requesting code.

Moreover, Preisler neither discloses nor suggests "receiving an identifier associated with the requested primitive function at the one of the set of system services from another one of the set of system services, thereby enabling the one of the set of system services to call the primitive function via the identifier associated with the requested primitive function." In other words, Preisler neither discloses nor suggests receiving an identifier of the requested primitive function replicating the first system service at the first system service from the second system service, enabling the first system service to call the primitive function using the identifier. In fact, in no manner does Preisler disclose a system service that receives an identifier of a primitive function that replicates the requesting system service. While a patch site is replaced by a branch to patch area instruction in Preisler, an identifier of the new section of code is not provided to the section of code being replaced. Accordingly, Applicant respectfully submits that claims 1 and 22 are patentable over Preisler.

With respect to claim 4, Preisler neither discloses nor suggests a method of providing replacement functions for a stack of system services, the stack of system services including one or more layers, each layer representing one of the system services, wherein lower layers provide services to upper layers in the stack, including "sending a primitive function request for a primitive function down from one of the layers of the stack of system services to another one of the layers in the stack of system services, the primitive function replicating the system service associated with the one of the layers in the stack." Specifically, Preisler neither discloses nor suggests a stack of system services in which one of the layers can communicate with another one of the layers. Moreover, Preisler neither discloses nor suggests sending a request from an upper layer in the stack to a lower layer in the stack for a primitive function replicating the system service associated with the upper layer in the stack. In addition, Preisler neither discloses nor suggests "when the another one of the layers is

responsible for performing at least one of input and output, returning a primitive function identifier associated with the primitive function to the one of the layers of the stack of system services." In other words, Preisler neither discloses nor suggests <u>returning a primitive function identifier identifying the requested primitive function to the upper layer in the stack when the lower layer receiving the request provides I/O functionality</u>. Accordingly, Applicant respectfully submits that claim 4 is patentable over Preisler.

With respect to claim 7, Preisler neither discloses nor suggests a method or apparatus for providing replacement functions for a stack of system services, the stack of system services including one or more layers, each layer representing one of the system services, wherein lower layers provide services to upper layers in the stack, including "sending a primitive function request for a primitive function down from a first one of the layers in the stack of system services to a second one of the layers in the stack of system services, the primitive function replicating the system service associated with the second one of the layers in the stack of system services in a manner such that implementation of the primitive function reduces or eliminates reliance on one or more system functions capable of becoming non-functional in the event of a system error." In other words, Preisler neither discloses nor suggests <u>sending a request from a first layer in the stack of system services to a second layer in the stack of system services, where the request is a request for a primitive function replicating the second service</u>. Similarly, Preisler neither discloses nor suggests "returning primitive function information associated with the primitive function to the first one of the layers." In addition, Preisler neither discloses nor suggests "storing the primitive function information to enable the first one of the layers in the stack of system services to communicate with the primitive function associated with the second one of the layers in the stack of system services." Accordingly, Applicant respectfully submits that claims 7 and 23 are patentable over Preisler.
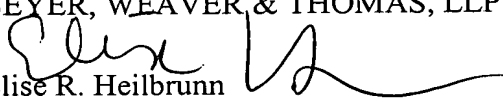
With respect to claim 13, Preisler neither discloses nor suggests a system for providing replacement system functions including a set of components providing a set of services and "a set of primitive functions associated with the set of services, the set of primitive functions replicating the set of services, wherein each of the set of primitive functions eliminates or reduces reliance on one or more system functions that are capable of becoming non-functional in the event of a system error." In addition, Preisler neither discloses nor suggests "a primitive function request mechanism adapted for returning one or more identifiers associated with one or more of the set of primitive functions." Accordingly,

Applicant respectfully submits that claim 13 is allowable.

As shown above, Preisler fails to teach each aspect of the claimed invention. Because Preisler fails to teach communication between two system services (e.g., layers in a stack of system services), sending a request from one system service to another system service for a primitive function (e.g., replicating the requesting or receiving system service), or receiving an identifier of a primitive function at one system service from another system service in response to such a request as set forth above, Preisler fails to teach all of the claim limitations as alleged by the Examiner. Accordingly, Applicants contend that the rejection is unsupported by the art and should be withdrawn. Thus, it is respectfully requested that the Examiner withdraw the rejection of claims under 35 USC §102.

If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for any extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 50-0388 (Order No. SUN1P376).

Respectfully submitted,
BEYER, WEAVER & THOMAS, LLP

Elise R. Heilbrunn
Reg. No. 42,649

BEYER, WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704
Tel. (510) 843-6200